Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: L2 aND (graphical ADJ debugger)

Display: 50 **Documents in Display Format:** REV **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Sunday, June 11, 2006 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=NO; OP=OR			
L3	L2 aND (graphical ADJ debugger)	2	L3
L2	717/124-131.ccls.	1420	L2
L1	loadmap	1	L1

END OF SEARCH HISTORY

Terms used **object debugger breakpoint** Found **6,025** of **177,263**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

21 The Mesa programming environment

Richard E. Sweet
June 1983 **ACM SIGPLAN Notices , ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 85 symposium on Language issues in programming environments**, Volume 18 , 20 Issue 6 , 7
Publisher: ACM Press
Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, index terms

People everywhere are developing multi-window, integrated programming environments for their favorite computers and languages. This paper describes the Mesa programming facilities of the Xerox Development Environment (XDE). It is interesting for several reasons. It has existed in something similar to its current form for about 5 years. It has more than 500 users, many interacting with it 8 or more hours a day. Several million lines of code have been written by these users, including large, ...

22 VAX DEBUG: an interactive, symbolic, multilingual debugger

Bert Beander
March 1983 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the symposium on High-level debugging SIGSOFT '83**, Volume 8 , 18 Issue 4 , 8
Publisher: ACM Press
Full text available: pdf(655.76 KB) Additional Information: full citation, abstract, references

Digital Equipment Corporation's VAX-11 Debugger, usually called VAX DEBUG or simply DEBUG, is an interactive, symbolic, and multilingual debugger which runs on the VAX-11 series of computers under the VMS operating system. The following gives an overview of VAX DEBUG and examines how it solves some of the problems inherent in the design of any such debugger. Particular attention is paid to how its command language is designed, how it distinguishes between addresses and values in command input, h ...

23 Multilingual debugging with the SWAT high-level debugger

James R. Cardell
March 1983 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the symposium on High-level debugging SIGSOFT '83**, Volume 8 , 18 Issue 4 , 8
Publisher: ACM Press
Full text available: pdf(794.24 KB) Additional Information: full citation, abstract, references

The SWAT (TM) debugger, Data General Corporation's Source Language Debugger, is an

interactive, high-level, symbolic debugging tool. It offers its users a full complement of standard high-level debugging features with a simple command format modeled after that of Data General's AOS and AOS/VS operating system Command Line Interpreter. Multilingual capability was a primary design goal and this has resulted in the benefits of both wide user acceptance and product extensibility.This paper presents ...

24 Experience with representing C++ program information in an object-oriented database

Tamiya Onodera

October 1994 **ACM SIGPLAN Notices , Proceedings of the ninth annual conference on Object-oriented programming systems, language, and applications OOPSLA '94**, Volume 29 Issue 10

Publisher: ACM Press

Full text available: pdf(1.26 MB) Additional Information: full citation, abstract, references, index terms

Two major issues related to storing program information in an OODB are sharing and clustering. The former is important since it prevents the database from consuming excessive disk space, while the latter is crucial, since it keeps clients running without thrashing. In our database, objects are shared across multiple programs' translation units, and are clustered by combining three techniques, namely, birth-order, death-order, and sharing-oriented clusterings. An initial experiment shows tha ...

25 Event and state-based debugging in TAU: a prototype

Sameer Shende, Janice Cuny, Lars Hansen, Joydip Kundu, Stephen McLaughry, Odile Wolf

January 1996 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools**

Publisher: ACM Press

Full text available: pdf(1.49 MB) Additional Information: full citation, references, index terms

26 Virtual images: interactive visualization of distributed object-oriented systems

Jean-Yves Vion-Dury, Miguel Santana

October 1994 **ACM SIGPLAN Notices , Proceedings of the ninth annual conference on Object-oriented programming systems, language, and applications OOPSLA '94**, Volume 29 Issue 10

Publisher: ACM Press

Full text available: pdf(1.91 MB) Additional Information: full citation, abstract, references, citings, index terms

In spite of growing needs in many areas, there is a lack of powerful graphical interfaces for interacting with large and complex sets of objects. Debugging, management and monitoring tools for object-oriented distributed systems or databases, for instance, need new interfaces that allow high quality visualization and interaction.We propose to use 3D interactive animations for representing large numbers of objects, complex relationships, and dynamic execution of concurrent activit ...

27 Debugging distributed object applications with the Eclipse platform

Giuliano Mega, Fabio Kon

October 2004 **Proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange**

Publisher: ACM Press

Full text available: pdf(244.98 KB) Additional Information: full citation, abstract, references

Debugging distributed applications is a well known challenge within the realm of Computer Science. Common problems faced by developers include: lack of an observable global state, lack of a central location from where to monitor possible states, non-

deterministic execution, heisenbugs, and many others. There are currently many good techniques available which could be employed in building a tool for circumventing some of those issues, especially when considering wide-spread middleware-induced mod ...

28 The design of an OCL query-based debugger for C++

Chanika Hobatr, Brian A. Malloy
March 2001 **Proceedings of the 2001 ACM symposium on Applied computing**
Publisher: ACM Press
Full text available: pdf(419.95 KB) Additional Information: full citation, references, index terms

Keywords: OpenC++, code generation, code instrumentation, debugging, meta-class, meta-object protocol, object constraint language, query, unified modelling language

29 An extensible object-oriented mixed-mod functional simulation system

Richard H. Lathrop, Robert S. Kirk
June 1985 **Proceedings of the 22nd ACM/IEEE conference on Design automation**
Publisher: ACM Press
Full text available: pdf(936.09 KB) Additional Information: full citation, abstract, references, citings, index terms

A LISP-based functional simulation system supporting a general concept of function and abstraction is described. SIMMER was developed primarily to support research into the relation of structure to function and associated description languages, and also to provide assistance to the designer analyzing especially difficult circuits. It consists of a general object-oriented message-passing functional simulator; a user-extensible intermediate-level base language for describing complex systems; ...

30 Query-based debugging of object-oriented programs

Raimondas Lencevicius, Urs Hölzle, Ambuj K. Singh
October 1997 **ACM SIGPLAN Notices , Proceedings of the 12th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '97**, Volume 32 Issue 10
Publisher: ACM Press
Full text available: pdf(1.84 MB) Additional Information: full citation, abstract, references, citings, index terms

Object relationships in modem software systems are becoming increasingly numerous and complex. Programmers who try to find violations of such relationships need new tools that allow them to explore objects in a large system more efficiently. Many existing debuggers present only a low-level, one-object-at-a-time view of objects and their relationships. We propose a new solution to overcome these problems: query-based debugging. The implementation of the query-based debugger described here offers ...

31 Debugging of globally optimized programs using data flow analysis

Roland Wismüller
June 1994 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1994 conference on Programming language design and implementation PLDI '94**, Volume 29 Issue 6
Publisher: ACM Press
Full text available: pdf(1.19 MB) Additional Information: full citation, abstract, references, citings, index terms

Advanced processor and machine architectures need optimizing compilers to be efficiently programmed in high level languages. Therefore the need for source level debuggers that

can handle optimized programs is rising. One difficulty in debugging optimized code arises from the problem to determine the values of source code variables. To ensure correct debugger behaviour with optimized programs, the debugger not only has to determine the variable's storage location or associated register. It m ...

32 The Mantis parallel debugger

Steven S. Lumetta, David E. Culler

January 1996 **Proceedings of the SIGMETRICS symposium on Parallel and distributed tools**

Publisher: ACM Press

Full text available: pdf(2.19 MB) Additional Information: full citation, references, index terms

33 Dynamic currency determination in optimized programs

D. M. Dhamdhere, K. V. Sankaranarayanan

November 1998 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 20 Issue 6

Publisher: ACM Press

Full text available: pdf(302.86 KB) Additional Information: full citation, abstract, references, citings, index terms

Compiler optimizations pose many problems to source-level debugging of an optimized program due to reordering, insertion, and deletion of code. On such problem is to determine whether the value of a varible is current at a breakpoint—that is, whether its actual value is the same as its expected value. We use the notion of dynamic currency of a variable in source-level debugging and propose the use of a minimal unrolled graph to reduce ...

Keywords: code instrumentation, code optimization, compiler, debugging optimized code, dynamic currency determination, dynamic slicing, minimal unrolled graph, source-level debugging

34 Debugging standard ML without reverse engineering

Andrew P. Tolmach, Andrew W. Appel

May 1990 **Proceedings of the 1990 ACM conference on LISP and functional programming**

Publisher: ACM Press

Full text available: pdf(1.29 MB) Additional Information: full citation, abstract, references, citings, index terms

We have built a novel and efficient replay debugger for our Standard ML compiler. Debugging facilities are provided by instrumenting the user's source code; this approach, made feasible by ML's safety property, is machine-independent and back-end independent. Replay is practical because ML is normally used functionally, and our compiler uses continuation-passing style; thus most of the program's state can be checkpointed quickly and compactly using call-with-current-continuation. Together, ...

35 Symbolic Debugging of Optimized Code

John Hennessy

July 1982 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 4 Issue 3

Publisher: ACM Press

Full text available: pdf(1.37 MB) Additional Information: full citation, references, citings, index terms

36 Debugging optimized code with dynamic deoptimization

Urs Hölzle, Craig Chambers, David Ungar

July 1992 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1992 conference on Programming language design and implementation PLDI '92**, Volume 27 Issue 7

Publisher: ACM Press

Full text available: pdf(1.26 MB) Additional Information: full citation, abstract, references, citings, index terms

SELF's debugging system provides complete source-level debugging (expected behavior) with globally optimized code. It shields the debugger from optimizations performed by the compiler by dynamically deoptimizing code on demand. Deoptimization only affects the procedure activations that are actively being debugged; all other code runs at full speed. Deoptimization requires the compiler to supply debugging information at discrete interrupt points

37 HyperWeb: a framework for hypermedia-based environments

James C. Ferrans, David W. Hurst, Michael A. Sennett, Burton M. Covnot, Wenguang Ji, Peter Kajka, Wei Ouyang

November 1992 **ACM SIGSOFT Software Engineering Notes , Proceedings of the fifth ACM SIGSOFT symposium on Software development environments SDE 5**, Volume 17 Issue 5

Publisher: ACM Press

Full text available: pdf(1.09 MB) Additional Information: full citation, abstract, references, citings, index terms

Software productivity and quality will increase as we improve our model of software and develop tools to support that model. Development environments must take into account that software is more than source; that it is more than text; and that it forms a highly interconnected web of information. Because more time is spent understanding and maintaining software than creating it, environments should strongly support browsing and reading. Finally, environments must be easy to customize. ...

38 Hardware assisted high level debugging: preliminary draft

W. Morven Gentleman, Henry Hoeksma

March 1983 **ACM SIGSOFT Software Engineering Notes , ACM SIGPLAN Notices , Proceedings of the symposium on High-level debugging SIGSOFT '83**, Volume 8 , 18 Issue 4 , 8

Publisher: ACM Press

Full text available: pdf(489.23 KB) Additional Information: full citation, abstract, references

Hardware assistance has long been used for logic level and functional unit level hardware debugging, as well as for machine language level software debugging. Such hardware assistance includes probes to detect signals, comparators to identify matches with expected patterns, buffers to record selected events, and independent logic and software to analyze and interpret the observed events. It can also include the ability to generate selected signals to stimulate the object being debugged and the a ...

39 A micro-debugger for Eclipse computers

Daniel Martin

April 1978 **Proceedings of the 16th annual Southeast regional conference**

Publisher: ACM Press

Full text available: pdf(210.31 KB) Additional Information: full citation, abstract, references

A micro-debugger for a Data General Eclipse S/200 computer is described. The Eclipse computer is a microprogrammable device with 56 bits/control word. No software for the WCS option is provided by the manufacturer. Since many microprograms contain errors which announce themselves only by causing the system to crash, and since the microprogrammer can access registers which are invisible to AL programmers, it is

difficult to locate errors in microprograms. The micro-debugger is a set of ALGOL, Ass ...

40 An object-oriented program development environment for the first programming course

Michael Kölling, John Rosenberg

March 1996 **ACM SIGCSE Bulletin , Proceedings of the twenty-seventh SIGCSE technical symposium on Computer science education SIGCSE '96**, Volume 28 Issue 1

Publisher: ACM Press

Full text available: pdf(834.44 KB) Additional Information: full citation, abstract, references, citings, index terms

Over the last ten years there has been a major shift in programming language design from procedural languages to object-oriented languages. Most universities have adopted an object-oriented language for their first programming course. However, far less consideration has been given to the program development environment. In this paper we argue that the environment is possibly more important than the language and existing environments fail to fully support the object-oriented paradigm. We describe ...

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next